A. Basic scheme functions: car, cdr, cons, append, lambda, if, cond, etc.
B. Map and Apply
C. Fold (I won’t ask you to define fold, though it isn’t hard)
D. Unrestricted lambdas
E. Representing data structures with lambdas
F. Representing data structures with lists
G. How Let's, Lambdas and Applications are evaluated.
H. Dynamic and Static binding
I. State and set! – if we get to them today.
J. The Y-combinator -- I won’t ask you to define it, but why do we care about it?
   I could ask for an unnamed definition of a recursive function. For example,
   (define c (lambda (f)
      (lambda (n)
         (cond
           [(zero? n) 1]
           [else (* n ((f f) (- n 1)))])))))
   Then (c c) is the factorial function

Unlike most CS exams for classes above 150, a lot of the exam will involve coding: Write a Scheme function to do thus and so. You should also be prepared for some conceptual questions: such as What does a lambda expression evaluate to, and why?